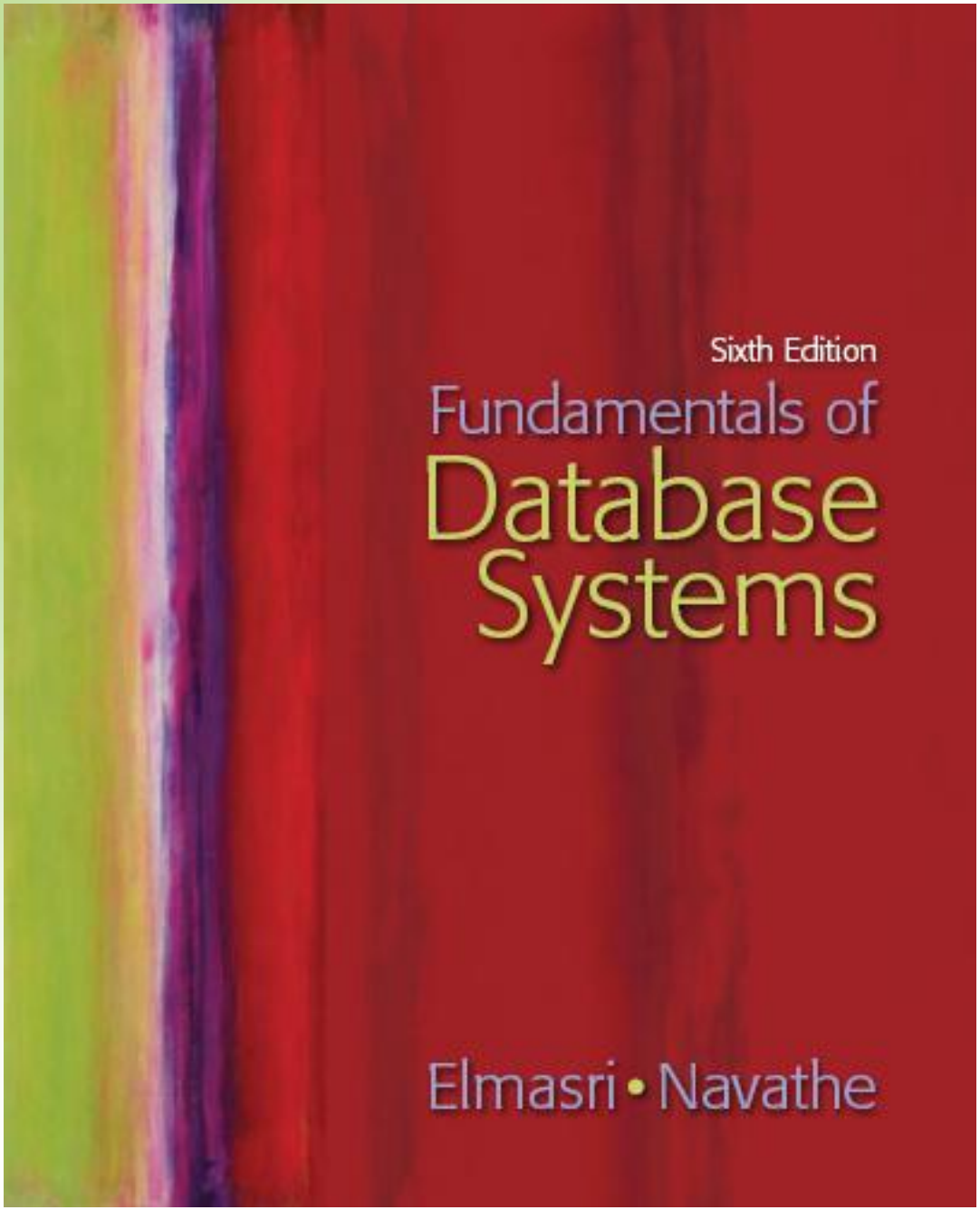
Chapter 10

Practical Database Design Methodology and Use of UML Diagrams



Addison-Wesley
is an imprint of
PEARSON

Chapter 10 Outline

- The Role of Information Systems in Organizations
- The Database Design and Implementation Process
- Use of UML Diagrams as an Aid to Database Design Specification
- Rational Rose: A UML-Based Design Tool
- Automated Database Design Tools

Practical Database Design Methodology and Use of UML Diagrams

- Design methodology
  - Target database managed by some type of database management system
- Various design methodologies
- **Large database**
  - Several dozen gigabytes of data and a schema with more than 30 or 40 distinct entity types

The Role of Information Systems in Organizations

- Organizational context for using database systems
  - Organizations have created the position of database administrator (DBA) and database administration departments
  - Information technology (IT) and information resource management (IRM) departments
    - Key to successful business management

The Role of Information Systems in Organizations (cont'd.)

- Database systems are integral components in computer-based information systems
- Personal computers and database system-like software products
  - Utilized by users who previously belonged to the category of casual and occasional database users
- **Personal databases** gaining popularity
- Databases are distributed over multiple computer systems
  - Better local control and faster local processing

The Role of Information Systems in Organizations (cont'd.)

- **Data dictionary systems** or **information repositories**
  - Mini DBMSs
  - Manage **meta-data**
- High-performance transaction processing systems require around-the-clock nonstop operation
  - Performance is critical

The Information System Life Cycle

- **Information system (IS)**
  - Resources involved in collection, management, use, and dissemination of information resources of organization

The Information System Life Cycle

- **Macro life cycle**
  - **Feasibility analysis**
  - **Requirements collection and analysis**
  - **Design**
  - **Implementation**
  - **Validation and acceptance testing**
  - **Requirements collection and analysis**

The Information System Life Cycle (cont'd.)

- The database application system life cycle: **micro life cycle**
  - System definition
  - Database design
  - Database implementation
  - Loading or data conversion

The Information System Life Cycle (cont'd.)

- Application conversion
- Testing and validation
- Operation
- Monitoring and maintenance

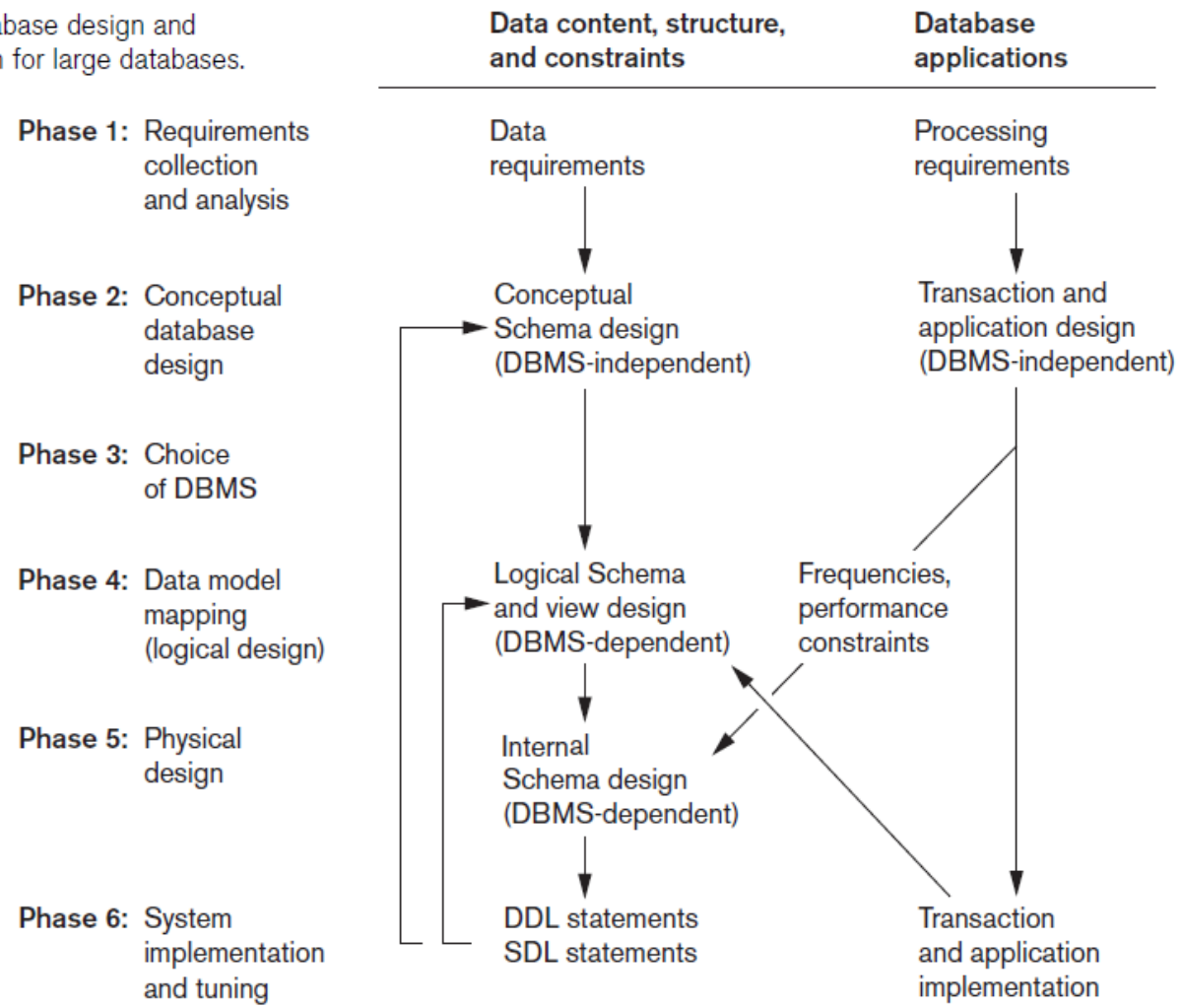
The Database Design and Implementation Process

- Design logical and physical structure of one or more databases
  - Accommodate the information needs of the users in an organization for a defined set of applications
- Goals of database design
  - Very hard to accomplish and measure
- Often begins with informal and incomplete requirements

The Database Design and Implementation Process (cont'd.)

- Main phases of the overall database design and implementation process:
  - 1. Requirements collection and analysis
  - 2. Conceptual database design
  - 3. Choice of a DBMS
  - 4. Data model mapping (also called logical database design)
  - 5. Physical database design
  - 6. Database system implementation and tuning

Figure 10.1
Phases of database design and implementation for large databases.



The Database Design and Implementation Process (cont'd.)

- Parallel activities
  - **Data content**, **structure**, and **constraints** of the database
  - Design of database applications
- **Data-driven** versus **process-driven** design
- **Feedback loops** among phases and within phases are common

The Database Design and Implementation Process (cont'd.)

- Heart of the database design process
  - **Conceptual database design (Phase 2)**
  - **Data model mapping (Phase 4)**
  - **Physical database design (Phase 5)**
  - **Database system implementation and tuning (Phase 6)**

Phase 1: Requirements Collection and Analysis

- Activities
  - Identify application areas and user groups
  - Study and analyze documentation
  - Study current operating environment
  - Collect written responses from users

Phase 1 (cont'd.)

- **Requirements specification techniques**
  - Oriented analysis (OOA)
  - Data flow diagrams (DFDs
  - Refinement of application goals
  - Computer-aided

Phase 2: Conceptual Database Design

- Phase 2a: Conceptual Schema Design
  - Important to use a conceptual high-level data model
  - Approaches to conceptual schema design
    - **Centralized (or one shot) schema design approach**
    - **View integration approach**

Phase 2: (cont'd.)

- Strategies for schema design
  - **Top-down strategy**
  - **Bottom-up strategy**
  - **Inside-out strategy**
  - **Mixed strategy**
- Schema (view) integration
  - Identify correspondences/conflicts among schemas:
    - **Naming conflicts, type conflicts, domain (value set) conflicts, conflicts among constraints**
  - Modify views to conform to one another
  - Merge of views and restructure

Phase 2: (cont'd.)

- Strategies for the view integration process
  - **Binary ladder integration**
  - **N-ary integration**
  - **Binary balanced strategy**
  - **Mixed strategy**
- Phase 2b: Transaction Design
  - In parallel with Phase 2a
  - Specify transactions at a conceptual level
  - Identify i**nput/output** and **functional behavior**
  - Notation for specifying processes

Phase 3: Choice of a DBMS

- Costs to consider
  - Software acquisition cost
  - Maintenance cost
  - Hardware acquisition cost
  - Database creation and conversion cost
  - Personnel cost
  - Training cost
  - Operating cost
- Consider DBMS portability among different types of hardware

Phase 4: Data Model Mapping (Logical Database Design)

- Create a conceptual schema and external schemas
  - In data model of selected DBMS
- Stages
  - System-independent mapping
  - Tailoring schemas to a specific DBMS

Phase 5: Physical Database Design

- Choose specific file storage structures and access paths for the database files
  - Achieve good performance
- Criteria used to guide choice of physical database design options:
  - Response time
  - Space utilization
  - Transaction throughput

Phase 6: Database System Implementation and Tuning

- Typically responsibility of the DBA
  - Compose DDL
  - Load database
  - Convert data from earlier systems
- Database programs implemented by application programmers
- Most systems include monitoring utility to collect performance statistics

Use of UML Diagrams as an Aid to Database Design Specification

- Use UML as a design specification standard
- Unified Modeling Language (UML) approach
  - Combines commonly accepted concepts from many object-oriented (O-O) methods and methodologies
  - Includes **use case diagrams**, **sequence diagrams**, and **statechart diagrams**

UML for Database Application Design

- Advantages of UML
  - Resulting models can be used to design relational, object-oriented, or object-relational databases
  - Brings traditional database modelers, analysts, and designers together with software application developers

Different Types of Diagrams in UML

- Structural diagrams
  - **Class diagrams and package diagrams**
  - **Object diagrams**
  - **Component diagrams**
  - **Deployment diagrams**

Different Types of Diagrams in UML (cont'd.)

- Behavioral diagrams
  - **Use case diagrams**
  - **Sequence diagrams**
  - **Collaboration diagrams**
  - **Statechart diagrams**
  - **Activity diagrams**

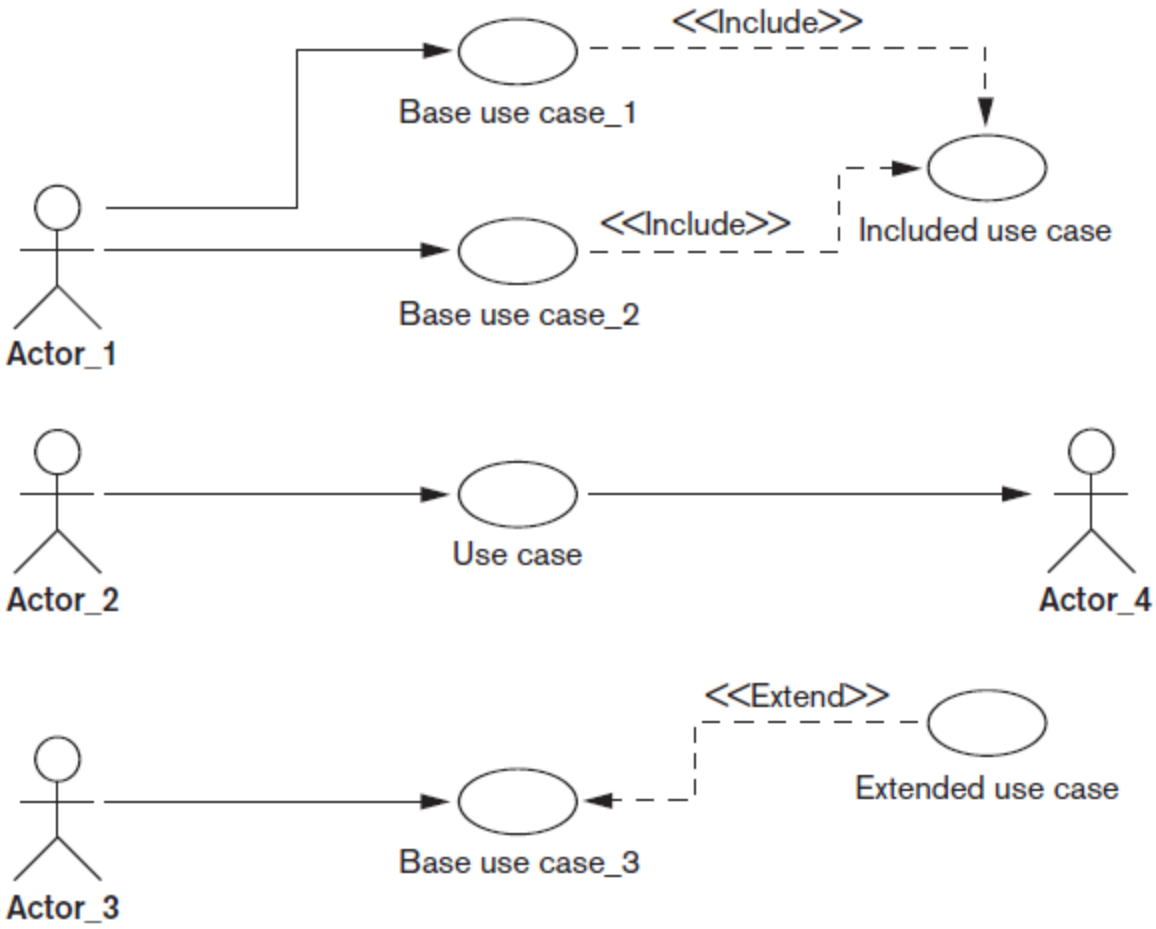


Figure 10.7
The use case diagram notation.

Different Types of Diagrams in UML (cont'd.)

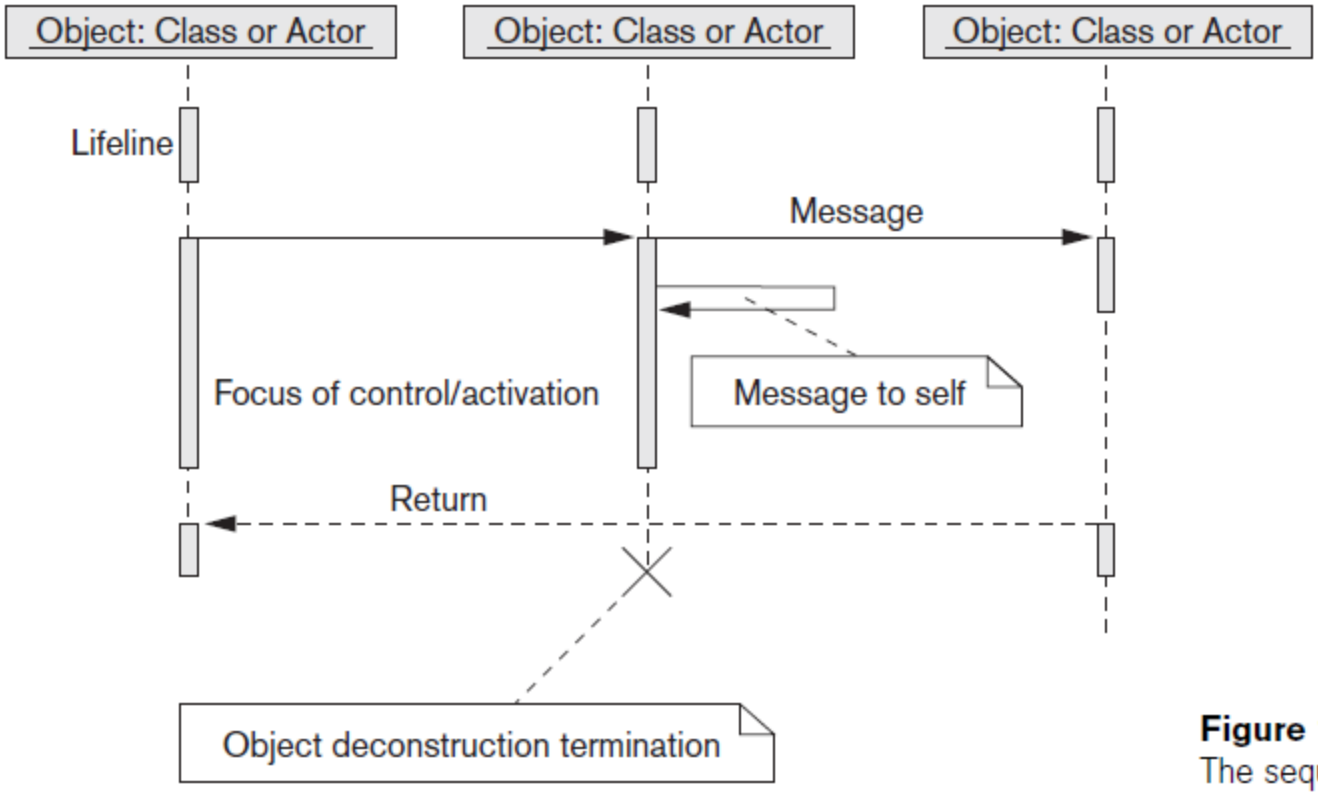


Figure 10.9
The sequence diagram notation.

Different Types of Diagrams in UML (cont'd.)

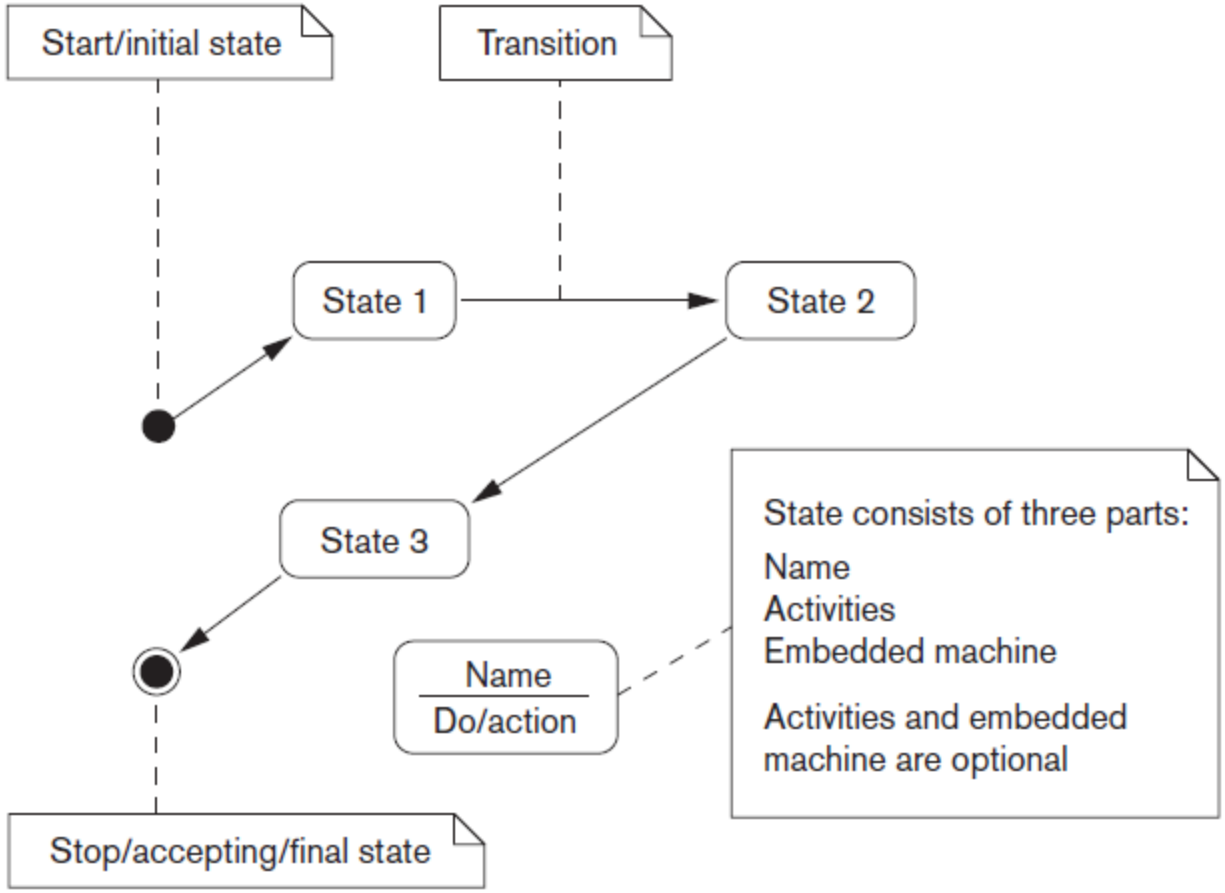


Figure 10.10
The statechart diagram notation.

Modeling and Design Example: UNIVERSITY Database

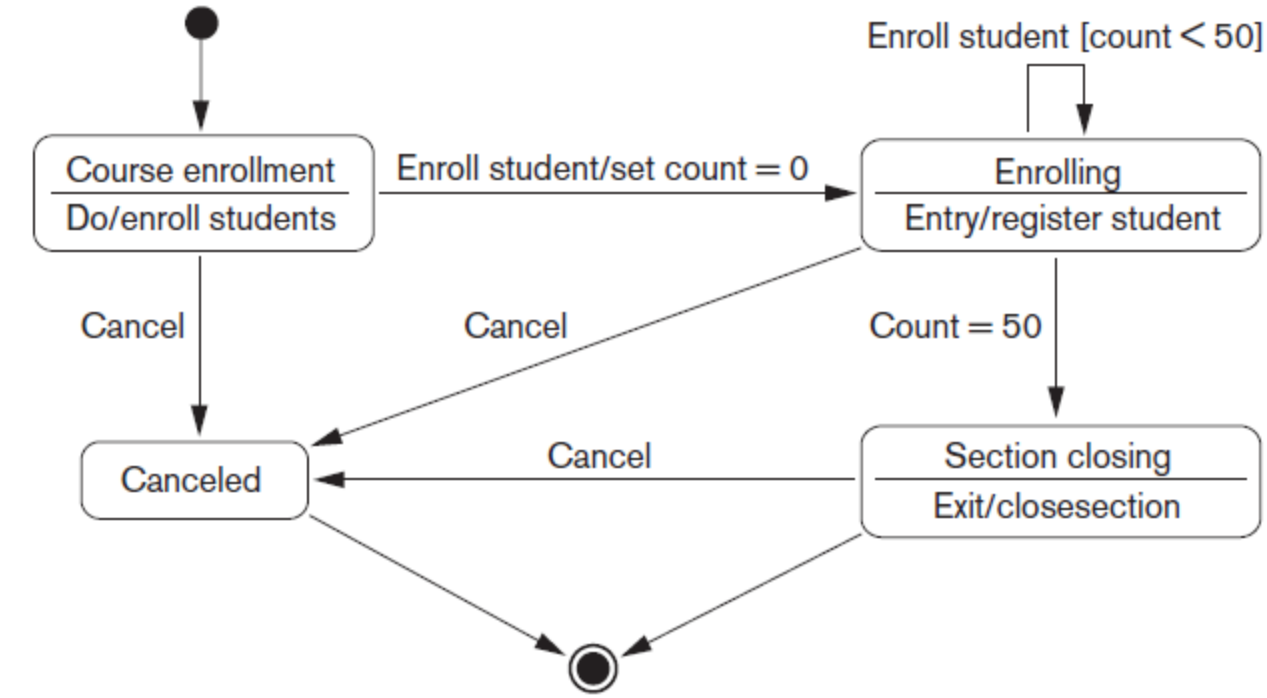


Figure 10.11
A sample statechart diagram for the UNIVERSITY database.

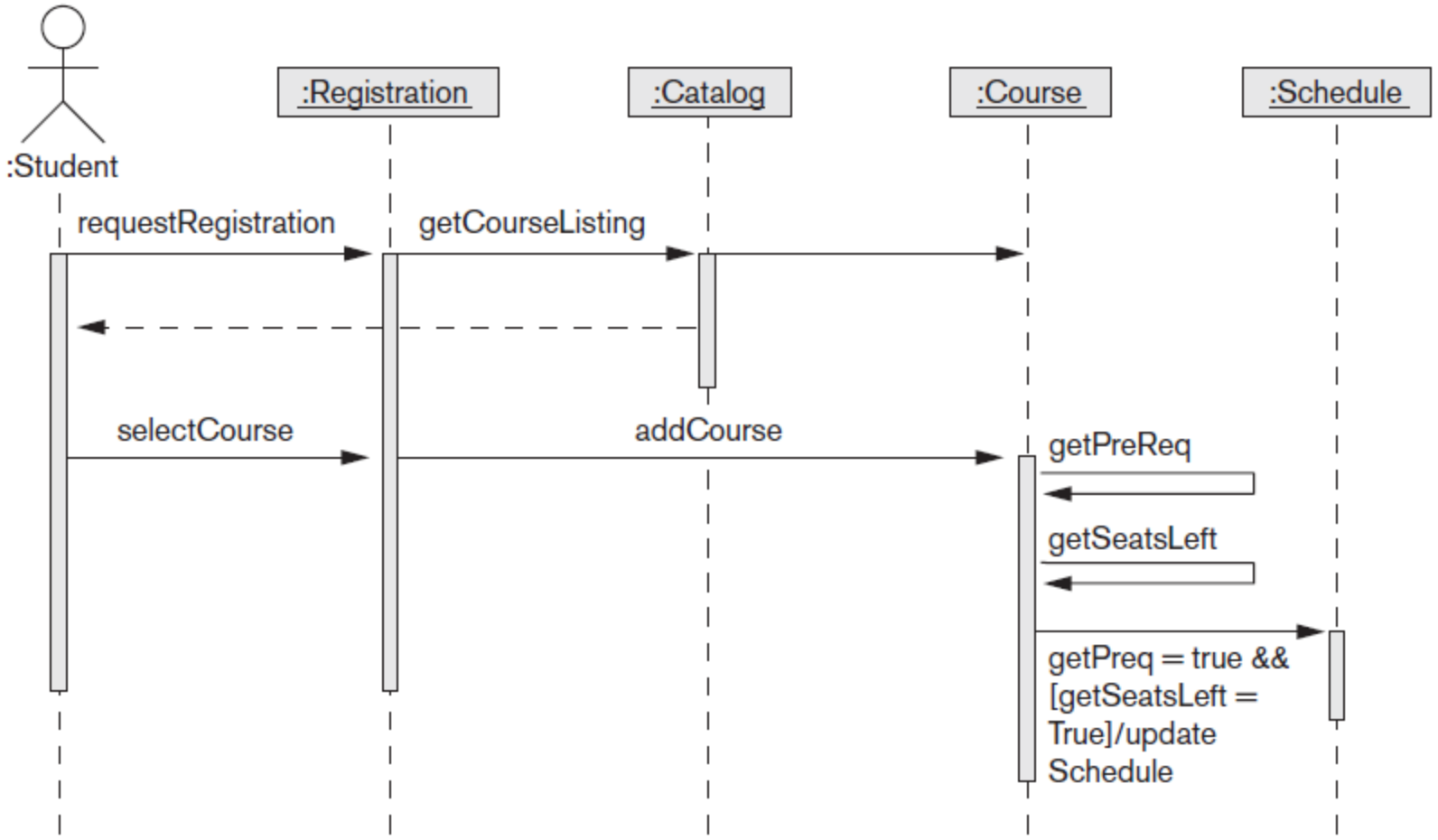


Figure 10.12
A sequence diagram for the UNIVERSITY database.

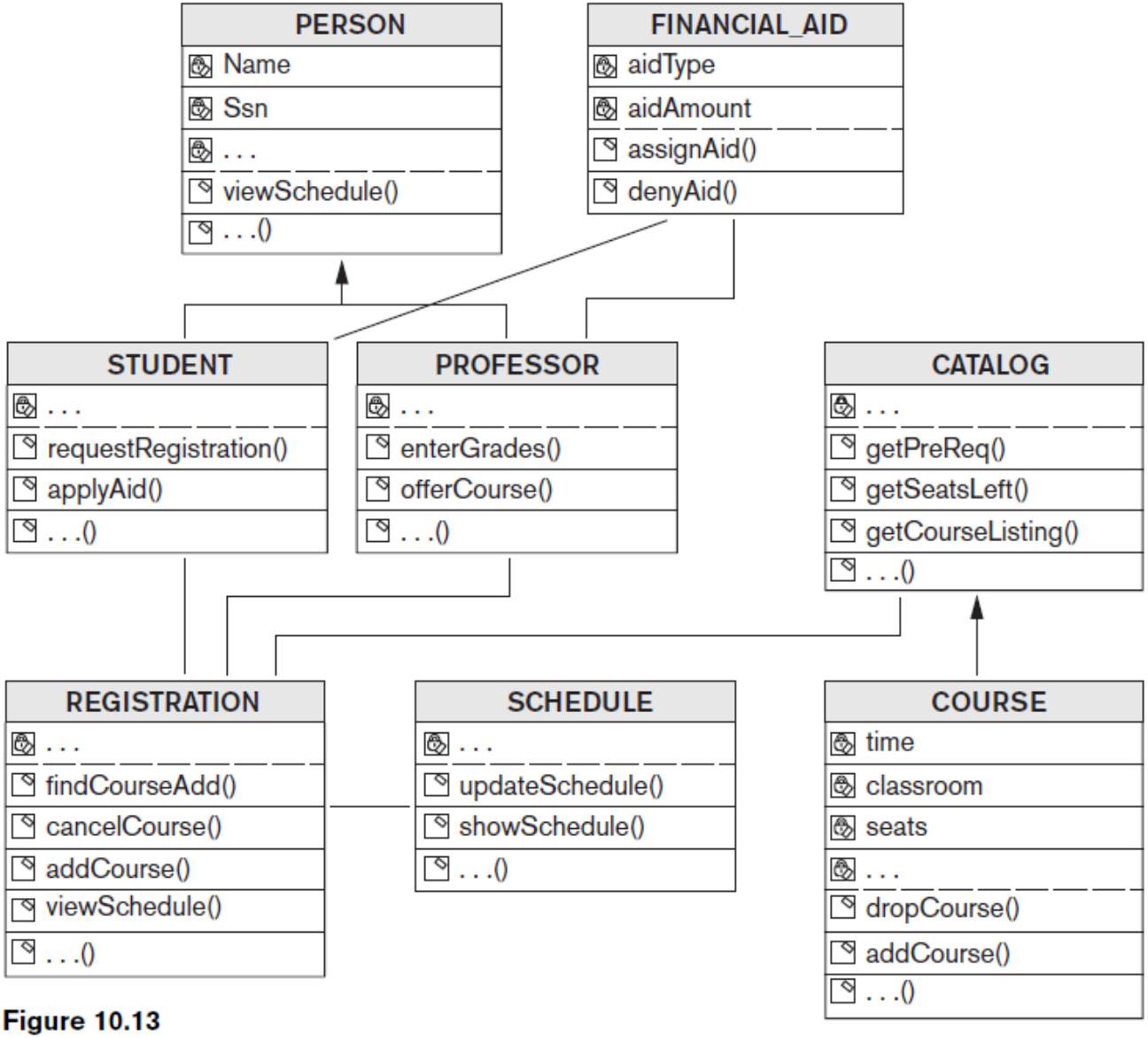


Figure 10.13
The design of the UNIVERSITY database as a class diagram.

Rational Rose: A UML-Based Design Tool

- Rational Rose for database design
  - Modeling tool used in the industry to develop information systems
- Rational Rose data modeler
  - Visual modeling tool for designing databases
  - Provides capability to:
    - **Forward engineer** a database
    - **Reverse engineer** an existing implemented database into conceptual design

Data Modeling Using Rational Rose Data Modeler

- Reverse engineering
  - Allows the user to create a conceptual data model based on an existing database schema specified in a DDL file
- Forward engineering and DDL generation
  - Create a data model directly from scratch in Rose
  - Generate DDL for a specific DBMS

Data Modeling Using Rational Rose Data Modeler (cont'd.)

- Conceptual design in UML notation
  - Build ER diagrams using class diagrams in Rational Rose
  - **Identifying relationships**
    - Object in a child class cannot exist without a corresponding parent object
  - **Non-identifying relationships**
    - Specify a regular association (relationship) between two independent classes

Data Modeling Using Rational Rose Data Modeler (cont'd.)

- Converting logical data model to object model and vice versa
  - Logical data model can be converted to an object model
  - Allows a deep understanding of relationships between conceptual and implementation models

Data Modeling Using Rational Rose Data Modeler (cont'd.)

- Synchronization between the conceptual design and the actual database
- Extensive domain support
  - Create a standard set of user-defined data types
- Easy communication among design teams
  - Application developer can access both the object and data models

Automated Database Design Tools

- Many CASE (computer-aided software engineering) tools for database design
- Combination of the following facilities
  - Diagramming
  - Model mapping
  - Design normalization

Automated Database Design Tools (cont'd.)

- Characteristics that a good design tool should possess:
  - Easy-to-use interface
  - Analytical components
  - Heuristic components
  - Trade-off analysis
  - Display of design results
  - Design verification

Automated Database Design Tools (cont'd.)

- Variety of products available
  - Some use expert system technology

Table 10.1 Some of the Currently Available Automated Database Design Tools

Company	Tool	Functionality
Embarcadero Technologies	ER/Studio DBArtisan	Database modeling in ER and IDEF1x Database administration and space and security management
Oracle	Developer 2000 and Designer 2000	Database modeling, application development
Persistence Inc.	PowerTier	Mapping from O-O to relational model
Platinum Technology (Computer Associates)	Platinum ModelMart, ERwin, BPwin, AllFusion Component Modeler	Data, process, and business component modeling
Popkin Software	Telelogic System Architect	Data modeling, object modeling, process modeling, structured analysis/design
Rational (IBM)	Rational Rose XDE Developer Plus	Modeling in UML and application generation in C++ and Java
Resolution Ltd.	XCase	Conceptual modeling up to code maintenance
Sybase	Enterprise Application Suite	Data modeling, business logic modeling
Visio	Visio Enterprise	Data modeling, design and reengineering Visual Basic and Visual C++

Summary

- Six phases of the design process
  - Commonly include conceptual design, logical design (data model mapping), physical design
- UML diagrams
  - Aid specification of database models and design
- Rational Rose and the Rose Data Modeler
  - Provide support for the conceptual design and logical design phases of database design